PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-065858
(43)Date of publication of application : 09.03.1999

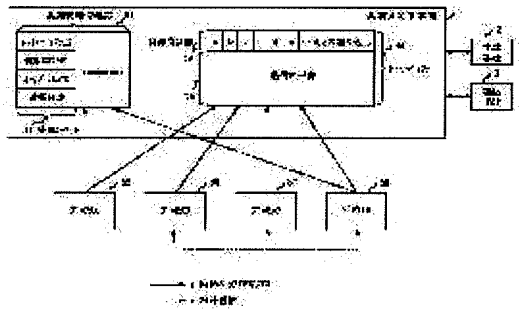
(51)Int.Cl. G06F 9/46

(21)Application number : 09-229895
(22)Date of filing : 26.08.1997

(71)Applicant : NEC AEROSPACE SYST LTD
(72)Inventor : KISHI TOSHIYUKI
OKABE KAZUTOSHI

(54) INTER-PROCESS COMMUNICATION SYSTEM USING COMMON MEMORY

(57)Abstract:
PROBLEM TO BE SOLVED: To easily perform an inter-process communication using the common memory at a high speed.
SOLUTION: A common management information part 31 is provided in the common memory space 1 and a communication source process A acquires a usable shuttle buffer ID therefrom and secure a shuttle buffer 32 with its ID. The shuttle buffer 32 has a control information part 33 which holds an owner process ID, etc., and a communication part 34 which holds communication data. A communication destination process B reads the contents of the shuttle buffer out on the basis of the ID and decreases the nonreference process number 33C of the control part of the shuttle buffer after completing reference. The shuttle buffer is released when the reference number 33C of only the control information part reaches 0.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－65858

(43)公開日　平成11年(1999)３月９日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/46	３４０	Ｇ０６Ｆ　9/46	３４０Ｂ ３４０Ｆ

審査請求　有　　請求項の数８　ＯＬ　（全８頁）

(21)出願番号　特願平9－229895

(22)出願日　平成９年(1997)８月26日

(71)出願人　000232221
日本電気航空宇宙システム株式会社
神奈川県横浜市港北区新横浜２丁目４番18号

(72)発明者　岸　俊行
神奈川県横浜市港北区新横浜二丁目４番18号　日本電気航空宇宙システム株式会社内

(72)発明者　岡部　一利
神奈川県横浜市港北区新横浜二丁目４番18号　日本電気航空宇宙システム株式会社内

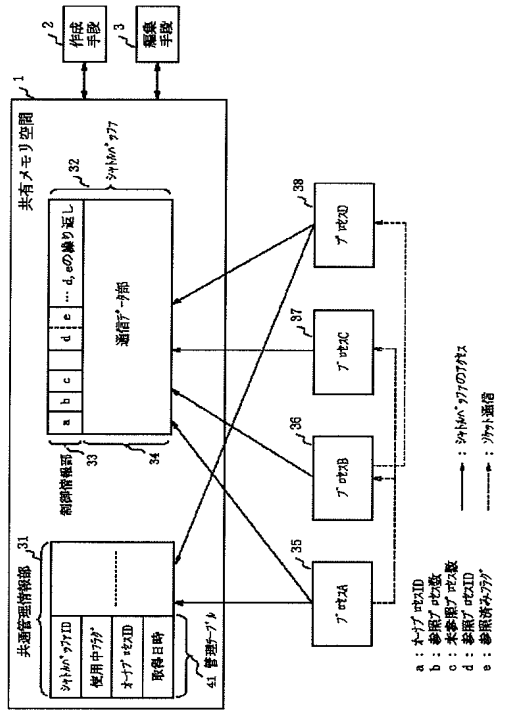
(74)代理人　弁理士　京本　直樹　（外２名）

(54)【発明の名称】　共有メモリを使用したプロセス間通信方式

(57)【要約】

【課題】共有メモリを使用したプロセス間通信を高速かつ容易に行う。

【解決手段】共有メモリ空間１に共通管理情報部３１を有し、通信元プロセスＡは、そこから使用可能なシャトルバッファＩＤを取得し、そのＩＤでシャトルバッファ３２を確保する。そのシャトルバッファ３２は、オーナプロセスＩＤ等を保持する制御情報部３３，並びに通信データを保持する通信部３４を有する。通信先プロセスＢは、そのＩＤを基にシャトルバッファの内容を読み出し、参照が完了したら当該シャトルバッファの制御部の未参照プロセス数３３Ｃをデクリメントする。シャトルバッファは、制御情報部のみ参照プロセス数３３Ｃが０になると解放される。



1

【特許請求の範囲】

【請求項１】　共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、前記制御情報部は、前記共有メモリ領域を獲得したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセスＩＤと、前記参照プロセスＩＤに示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含み、前記制御情報部を参照し、変更する編集手段と、前記編集手段において、前記未参照プロセス数の値から前記共有メモリ領域を参照するプロセスがなくなったことを検出した場合は前記共有メモリ領域を解放する解放手段とを有することを特徴とする共有メモリを使用したプロセス間通信方式。

【請求項２】　前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備えたことを特徴とする請求項１記載の共有メモリを使用したプロセス間通信方式。

【請求項３】　前記共通管理情報部は、前記共有メモリ領域の１つ毎に対応した管理テーブルを複数持つことを特徴とする請求項２記載の共有メモリを使用したプロセス間通信方式。

【請求項４】　前記編集手段において、前記参照プロセス数を変更し、前記参照プロセスＩＤと前記参照済みフラグを設定することにより、前記共有メモリ領域に設定された前記通信データ部の内容を他のプロセスに受け渡すことを特徴とする請求項１，２，または３記載の共有メモリを使用したプロセス間通信方式。

【請求項５】　共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、通信元プロセスは、前記制御情報部にプロセス間の制御のための制御情報を通信データ部に通信するためのデータをそれぞれ設定し、使用している前記共有メモリ領域の場所を第１の通信先プロセスに通知する第１の通知手段と、前記第１の通知手段によって通知された前記第１の通信先プロセスは、前記共有メモリ領域を参照し、受信した前記通信データを第２のプロセスに通知する必要がある場合は、既に設定された前記制御情報を変更し第２の通信先プロセスのＩＤを前記制御情報部に新たに設定した後、前記第２の通信先プロセスに通知する第２の通知手段と、第２の通知手段によって通知された前記第２の通信先プロセスは、前記第１の通知手段によって通知された前記第１のプロセスと同じ動作を行う動作手段と、前記第１の通知手段、前記第２の通知手段、または前記動作手段によって通知された全てのプロセスの前記共有メモリ領域への参照が完了した時点で、前記共有メモリ領域を解放する解放手段とを有することを特徴とする共有メモリを使用したプロセス間通信方式。

2

【請求項６】　前記制御情報部は前記共有メモリ領域を獲得したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセスＩＤと、前記参照プロセスＩＤに示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含むことを特徴とする請求項５記載の共有メモリを使用したプロセス間通信方式。

【請求項７】　前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備え、前記共通管理情報部は前記共有メモリ領域の１つ毎に対応した管理テーブルを複数持つことを特徴とする請求項５または６記載の共有メモリを使用したプロセス間通信方式。

【請求項８】　前記管理テーブルは、前記共有メモリ空間での前記共有メモリ領域の場所の識別を示す共有メモリ領域ＩＤと、前記共有メモリ領域の使用の有無を示す使用中フラグと、前記共有メモリ領域を確保したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を獲得した日時を示す取得日時とを含むことを特徴とする請求項３または７記載の共有メモリを使用したプロセス間通信方式。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、コンピュータプログラムにおけるプロセス間の通信に関し、特に共有メモリを使用したプロセス間通信に関する。

【０００２】

【従来の技術】一般的にプロセス間の通信で大量のデータをやり取りする場合、通信先が同一のコンピュータであるならば共有メモリを使用する方法が最も性能が良い。

【０００３】しかし、複数のプロセスに同じデータを受け渡したり、更に、受け取ったプロセスがまた別のプロセスにそのデータを受け渡したりしようとすると、受け渡し用に取得した共有メモリ領域を解放するための制御が複雑になるばかりでなく、そのための通信が頻繁に発生し、システムの性能を低下させる。また、共有メモリの取得・解放を単純にするために、予め取得した共有メモリ領域を使い回してプロセス間通信を行おうとした場合、今度は、受け渡し先のプロセスの参照が終わるまで次の書込が待たされたり、受け渡し先の全てのプロセスの参照が完了したことを通信元のプロセスに知られるための通信が必要になる。

【０００４】例えば、特開平７－１２９４１６公報に開

示されているように、通信元プロセスが共有メモリの取得、および解放を行っていた。従って、通信先プロセス全てが参照し終わった後に、通信元プロセスに共有メモリの解放要求を通知する必要があった。

【０００５】また、特開平４－３６４５５０公報に開示されているように、共有メモリに管理情報を設定し、通信元プロセスが共有メモリの解放を意識せずに済んでいるが、通信先プロセスが通信元プロセスの管理情報を操作するので、複数のプロセスに同時に同じ領域（共有メモリ）を使用してデータを送信することは出来ない。すなわち、１対多数の通信は考慮されていなかった。

【０００６】

【発明が解決しようとする課題】第１の問題点は、特開平７－１２９４１６公報のように、通信元プロセスがプロセス間通信で使用済みとなった共有メモリを削除する場合、共有メモリの参照完了を通知するため通信を行う必要があり、プログラム（特に、通信元プロセス）の制御が複雑になるばかりではなく、通信量の増大、並びに通信元プロセスの負荷増大によるシステム性能の低下が発生することである。

【０００７】第２の問題点は、特開平４－３６４５５０公報のように、通信先プロセスが共有メモリ上の制御情報を操作することによりその領域を未使用状態にする場合、同じ共有メモリ領域を使用して複数のプロセスにデータを受け渡すことができないことである。その理由は、通信先のプロセスのどれか一つが制御情報を操作してしまうとその共有メモリ領域が未使用の状態になり、次の通信に使われてしまうからである。

【０００８】本発明の目的は、共有メモリを使用したプロセス間通信を高速かつ容易に行うことにある。

【０００９】本発明の他の目的は、１対多数の通信を可能にすることにある。

【００１０】本発明の他の目的は、使用者に意識させない構造とし、プロセス間通信の生産性の向上を高めることにある。

【００１１】

【課題を解決するための手段】本発明の共有メモリを使用したプロセス間通信方式は、共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、前記制御情報部は、前記共有メモリ領域を獲得したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセスＩＤと、前記参照プロセスＩＤに示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含み、前記制御情報部を参照し、変更する編集手段と、前記編集手段において、前記未参照プロセス数の値から前記共有メモリ領域を参照するプロセスがなくなったことを検出した場合は前記共有メモリ領域を解放する解放手段とを有することを特徴としている。

【００１２】上記の他に、前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備えたことを特徴としている。

【００１３】前記共通管理情報部は、前記共有メモリ領域の１つ毎に対応した管理テーブルを複数持つことを特徴としている。

【００１４】前記編集手段において、前記参照プロセス数を変更し、前記参照プロセスＩＤと前記参照済みフラグを設定することにより、前記共有メモリ領域に設定された前記通信データ部の内容を他のプロセスに受け渡すことを特徴としている。

【００１５】また、本発明の共有メモリを使用したプロセス間通信方式は、共有メモリ空間を利用したプロセス間通信方式において、通信に使用する前記共有メモリ空間に制御情報部と通信データ部とを付加した共有メモリ領域を備え、通信元プロセスは、前記制御情報部にプロセス間の制御のための制御情報を通信データ部に通信するためのデータをそれぞれ設定し、使用している前記共有メモリ領域の場所を第１の通信先プロセスに通知する第１の通知手段と、前記第１の通知手段によって通知された前記第１の通信先プロセスは、前記共有メモリ領域を参照し、受信した前記通信データを第２のプロセスに通知する必要がある場合は、既に設定された前記制御情報を変更し第２の通信先プロセスのＩＤを前記制御情報部に新たに設定した後、前記第２の通信先プロセスに通知する第２の通知手段と、前記第１の通知手段、前記第２の通知手段、またはによって通知された全てのプロセスの前記共有メモリ領域への参照が完了した時点で、前記共有メモリ領域を解放する解放手段とを有することを特徴としている。

【００１６】前記制御情報部は前記共有メモリ領域を獲得したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を参照するプロセスの数を示す参照プロセス数と、前記共有メモリ領域を参照中またはまだ参照していないプロセスの数を示す未参照プロセス数と、前記共有メモリ領域を参照するプロセスの識別を示す参照プロセスＩＤと、前記参照プロセスＩＤに示されたプロセスが前記共有メモリ領域を参照したのかどうかを示す参照済みフラグとを含むことを特徴としている。

【００１７】前記共有メモリ空間に前記共有メモリ領域の使用状況を管理する共通管理情報部を備え、前記共通管理情報部は前記共有メモリ領域の１つ毎に対応した管理テーブルを複数持つことを特徴としている。

【００１８】前記管理テーブルは、前記共有メモリ空間の前記共有メモリ領域の場所の識別を示す共有メモリ領域ＩＤと、前記共有メモリ領域の使用の有無を示す使用

5

中フラグと、前記共有メモリ領域を確保したプロセスの識別を示すオーナプロセスＩＤと、前記共有メモリ領域を獲得した日時を示す取得日時とを含むことを特徴としている。

【００１９】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。

【００２０】図３は、本発明の共有メモリを使用したプロセス間通信方式を適用したシステムの構成図である。

【００２１】図３を参照すると、本発明の実施の形態は、共有メモリ空間１と、共有メモリ空間１内に共有メモリ領域（以降、シャトルバッファと呼ぶ）であるシャトルバッファ３２と、プロセスＡ３５と、プロセスＢ３６と、プロセスＣ３７と、プロセスＤ３８と、シャトルバッファ３２を使用するためにプロセスＡ３５，Ｂ３６，Ｃ３７，およびＤ３８が共有する共有メモリ空間１内にシャトルバッファを管理するための共通管理情報部３１と、シャトルバッファ３２の作成を行う作成手段２と、シャトルバッファ３２の編集（参照、変更、または解放）を行う編集手段３を備えている。

【００２２】共通管理情報部３１には、複数の管理テーブル４１が存在し、管理テーブル４１は１つのシャトルバッファを管理するために１つ存在する。図３に示す通り共通管理情報部３１の管理テーブル４１のフォーマットとして、シャトルバッファＩＤと、使用中フラグと、オーナプロセスＩＤと、シャトルバッファの取得日時とが存在する。シャトルバッファＩＤは共有メモリ空間１に存在するシャトルバッファのアドレス位置に対応している。使用中フラグはシャトルバッファを使用しているか使用していないかを示し、「使用中」または「未使用」で表示される。オーナプロセスＩＤにはシャトルバッファを獲得したプロセスＩＤが設定される。取得日時にはシャトルバッファを獲得した日時が設定される。

【００２３】システムの環境設定時に、システム全体で使用できるシャトルバッファの最大件数分、共有メモリ空間１に共通管理情報部３１（管理テーブル４１を最大件数分の領域）を用意する。

【００２４】図３のシャトルバッファ３２は、通信を行おうとするプロセスＡ３５が、共有メモリ空間１上に取得した領域であり、これは、制御情報部３３，および通信データ部３４から構成される。制御情報部３３のフォーマットの詳細は以下の通りである。

【００２５】ａは、オーナプロセスＩＤを示し、シャトルバッファ３２を取得したプロセスＩＤで、獲得したプロセスのみシャトルバッファ３２の通信データ部３４への書き込みが行える。

【００２６】ｂは、参照プロセス数を示し、シャトルバッファ３２を参照するプロセスの数を意味する。

【００２７】ｃは、未参照プロセス数を示し、シャトルバッファ３２を参照中、またはまだ参照していないプロセスの数を意味する。

6

【００２８】ｄは、参照プロセスＩＤを示し、シャトルバッファ３２を参照するプロセスのＩＤを意味する。

【００２９】ｅは、参照済みフラグを示し、参照プロセスＩＤに表示されたプロセスについて、シャトルバッファ３２を参照したのかどうかを示すフラグを意味している。。

【００３０】図１は、図３の作成手段２の動作をフローチャートで示したもので、シャトルバッファ作成手順を示す。

【００３１】図２は、図３の編集手段３の動作をフローチャートで示したもので、シャトルバッファの参照、変更、または解放手順を示す。

【００３２】次に、図１，図２，および図３を参照して、本発明の実施の形態の動作について説明する。

【００３３】先ず、図１と図２を参照して、シャトルバッファの作成手順にについて説明する。

【００３４】今、プロセスＡ３５の処理を行うと、プロセスＡ３５はプロセスＢ３６，プロセスＣ３７の処理をそれぞれ必要とし、プロセスＢ３６の処理を行うと、プロセスＢ３６はプロセスＤ３８の処理を必要とする。

【００３５】シャトルバッファを用いて通信を行おうとするプロセスＡ３５は、共通管理情報部３１から使用可能なシャトルバッファＩＤ（管理テーブル４１の使用中フラグが「未使用」であるもの）を検索する（図１のステップＳ１１，Ｓ１２）。シャトルバッファの空きがない場合は、空きがでるまで待たされる。

【００３６】使用可能なシャトルバッファＩＤを見つけたら、共通管理情報部３１の管理テーブル４１において、シャトルバッファＩＤに対応する使用中フラグを「使用中」に、自プロセスＩＤをオーナプロセスＩＤに、システム日時を取得日時に設定する（ステップＳ１３）。

【００３７】次に、管理テーブル４１のシャトルバッファＩＤから共有メモリ空間１上にシャトルバッファ３２の領域を取得し、自プロセスＩＤをシャトルバッファのオーナプロセスＩＤに、通信先のプロセスＩＤ（プロセスＢ３６，プロセスＣ３７）を参照プロセスＩＤに、通信先のプロセス数を参照プロセス数に設定する。未参照プロセス数は、参照プロセス数と同じ値を設定する（最初は全て未参照なので、未参照プロセス数＝参照プロセス数＝２となる）。また、各参照プロセスの参照済みフラグは全て「未参照」に設定する（ステップＳ１４）。

【００３８】次に、プロセスＡ３５はシャトルバッファ３２の通信データ部３４に通信データを設定し（ステップＳ１５）、プロセスＢ３６およびプロセスＣ３７にソケット通信によりシャトルバッファＩＤを通知する（ステップＳ１６）。

【００３９】上記手順により、シャトルバッファが作成され、通信先プロセスにデータが受け渡される。

【００４０】次に、図２と図３を参照して、シャトルバッファ参照、変更または解放手順について説明する。

【００４１】前記手順により、プロセスＡ３５からソケット通信によりシャトルバッファＩＤを通知されたプロセスＢ３６およびプロセスＣ３７は、通知されたシャトルバッファＩＤからシャトルバッファを参照する（図２のステップＳ２１）。

【００４２】プロセスＢ３６は、プロセスＤ３８にシャトルバッファ３２の内容を受け渡すため、シャトルバッファ３２の参照プロセス数および未参照プロセス数に１加算し（未参照プロセス数＝３）、プロセスＤ３８のプロセスＩＤをシャトルバッファ３２の参照プロセスＩＤに追加してその参照済みフラグを「未参照」に設定し、そしてプロセスＤ３８にソケット通信でシャトルバッファＩＤを通知する（ステップＳ２２，Ｓ２３）。

【００４３】次に、プロセスＢ３６はシャトルバッファ３２の自プロセスの参照済みフラグを「参照済み」に、未参照プロセス数を－１する（未参照プロセス＝２）（ステップＳ２４）。

【００４４】ここで、シャトルバッファ３２の未参照プロセス数をチェックするが（ステップＳ２５）、未参照プロセスは０でないので何もしない。

【００４５】プロセスＣ３７も同様にシャトルバッファを参照するが、他に通知するプロセスがないので、参照済みフラグの変更および未参照プロセス数のデクリメントのみ行う（未参照プロセス数＝１）。ここでもシャトルバッファ３２の未参照プロセス数をチェックするが、未参照プロセス数は０でないので何もしない。

【００４６】プロセスＢ３６から通知を受けたプロセスＤ３８はシャトルバッファを参照後、参照済みフラグの変更および未参照プロセス数のデクリメントを行う（未参照プロセス数＝０）。ここで、未参照プロセス数が０になるので、シャトルバッファ３２を解放して、共通管理情報部３１のシャトルバッファＩＤの使用中フラグを「未使用」に、オーナプロセスＩＤをクリア、取得日時をクリアする（ステップＳ２６，Ｓ２７）。

【００４７】上記説明において、共通管理情報部３１の管理テーブル４１とシャトルバッファ３２の状態遷移図を描くと図４の通りになる。

【００４８】図４において、図４－（１）は、プロセスＡ（ＩＤ＝０００１とする）がシャトルバッファ（ＩＤ＝５００１とする）を取得（作成）し、プロセスＢ（ＩＤ＝０００２とする）とプロセスＣ（ＩＤ＝０００３とする）に通知するときの状態遷移図である。図４－（２）は、プロセスＢ（ＩＤ＝０００２）が内容を参照後、プロセスＤ（ＩＤ＝０００４とする）に通知する状態遷移図である。図４－（３）は、プロセスＣ（ＩＤ＝０００３）とプロセスＤ（ＩＤ＝０００４）が内容を参照後シャトルバッファ（ＩＤ＝５００１）を解放した状態遷移図である。図４－（４）は、プロセスＢ，Ｃ，およびＤの全ての参照が終わり、シャトルバッファ（ＩＤ＝５００１）を解放した状態遷移図である。

【００４９】上記説明では、シャトルバッファＩＤを通知するためのプロセス間通信をソケット通信により行っているが、ソケット通信以外の手段を用いても可能なことは言うまでもない。また、図３ではプロセス数を４つにしてあるが、プロセス数がいくつであってもよいことはもちろんである。

【００５０】以上説明したように、複数プロセスへの大量データのブロードキャスト通信、並びに受信データの他プロセスへの引き渡しを、高速、かつ容易に実現することができる。

【００５１】また、全てのプロセスの参照が完了したシャトルバッファは、最後に参照を完了したプロセスが解放するので、プログラムの制御を簡単にすると共に、通信量の増大によるシステム性能の低下を防ぐことができる。

【００５２】更に、多量の通信データをシャトルバッファを使用することにより、同じ領域（シャトルバッファ）で複数のプロセスに通知することができるため１対多数の通信ができる。

【００５３】更に、通信用共有メモリ生成後の参照、譲渡、および解放の管理をシャトルバッファを使用することにより使用者に意識させない構造となるため、プロセス間通信の生産性が高まる。

【００５４】更に、シャトルバッファと、共通管理情報部を設けることにより、管理情報を全て管理することとなり、状態遷移図でも示したように共有メモリの作成後の流れを容易にトレースすることができ、システム稼働後の不具合等の調査を容易に行えることができる。

【００５５】なお、上記説明において、ソケット通信のソケットとは、ＯＳＩ参照モデルのＴＣＰ／ＩＰネットワーク内のトランスポート層上でのプロセス間通信によるデータ送受信を行うアプリケーションプログラムを作成するためのＡＰＩ（アプリケーション・プログラミング・インタフェース）であり、ソケット通信とは、ソケットを利用した通信のことである。

【００５６】

【発明の効果】本発明の共有メモリを使用したプロセス間通信方式は、前述の通り以下の効果をもたらす。

【００５７】本発明の効果は、複数プロセスへの大量データのブロードキャスト通信、並びに受信データの他プロセスへの引き渡しを、高速、かつ容易に実現することできるということである。これにより、サーバから複数クライアントへの通信、並びに他プロセスへのデレゲーション（作業委託）通信を行うシステムを高性能かつ高生産で開発することできる。これは、本発明が、１つの共有メモリ領域を用いたプロセス間通信をメモリ管理情報を交換するための通信を行うことなく実現する方式を提供するからである。

9

【図面の簡単な説明】

【図１】本発明の共有メモリを使用したプロセス間通信方式のシャトルバッファ作成手順を示すフローチャートである。

【図２】本発明の共有メモリを使用したプロセス間通信方式のシャトルバッファ参照、変更または解放手順を示すフローチャートである。

【図３】本発明の共有メモリを使用したプロセス間通信方式を用いたシステムのブロック図である。

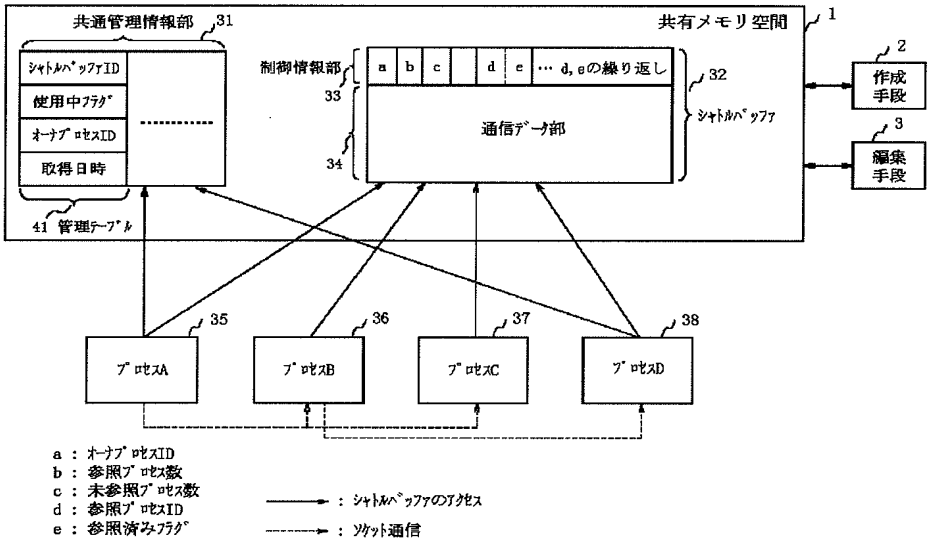
【図４】本発明の共通管理情報部とシャトルバッファの関係を示した状態遷移図である。

【符号の説明】

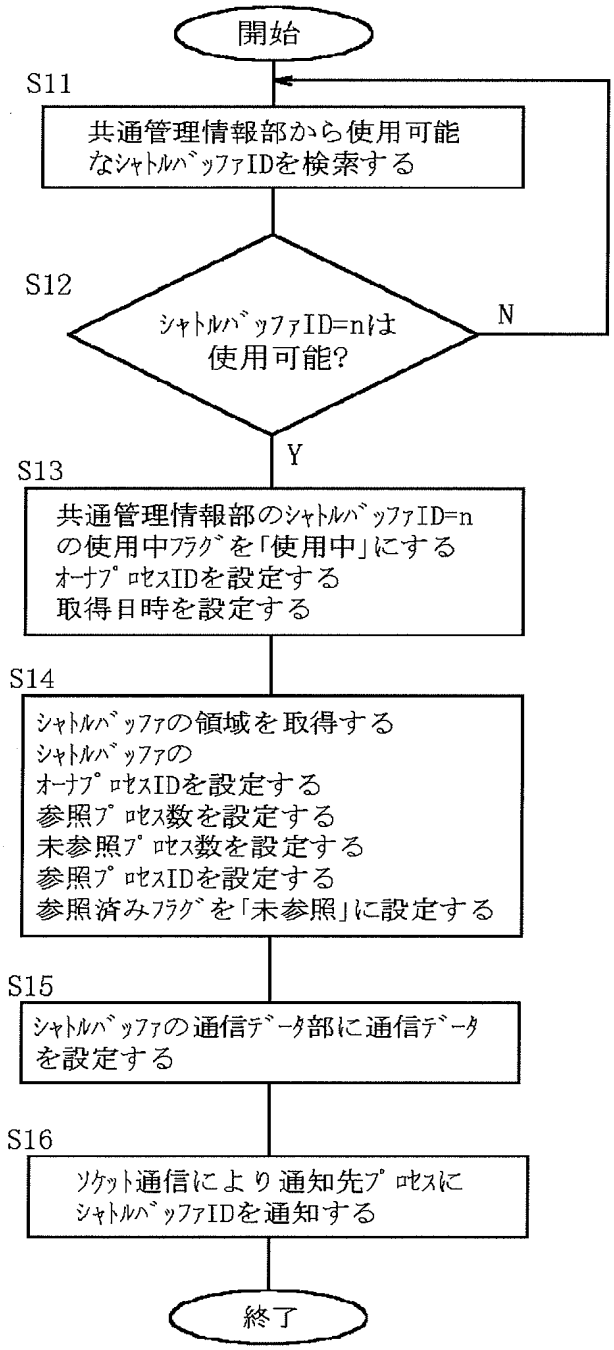
10

1	共有メモリ空間
2	作成手段
3	編集手段
３１	共通管理情報部
３２	シャトルバッファ
３３	制御情報部
３４	通信データ部
３５	プロセスＡ
３６	プロセスＢ
３７	プロセスＣ
３８	プロセスＤ
４１	管理テーブル

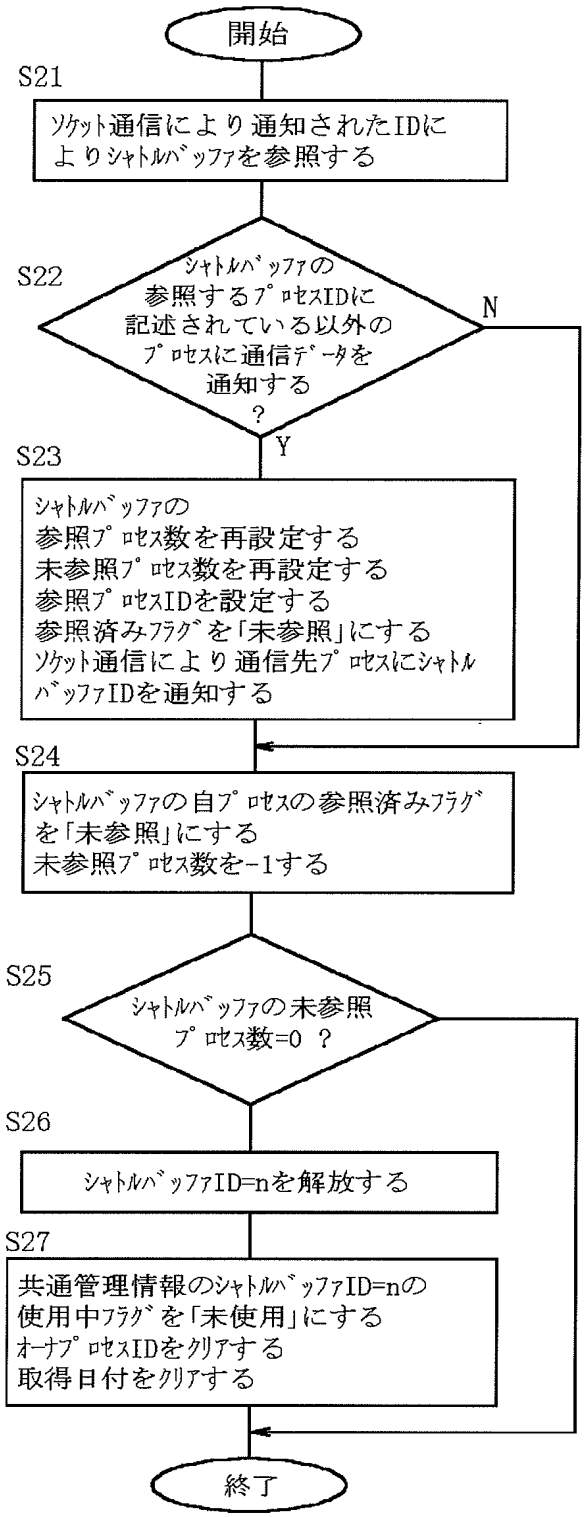
【図３】



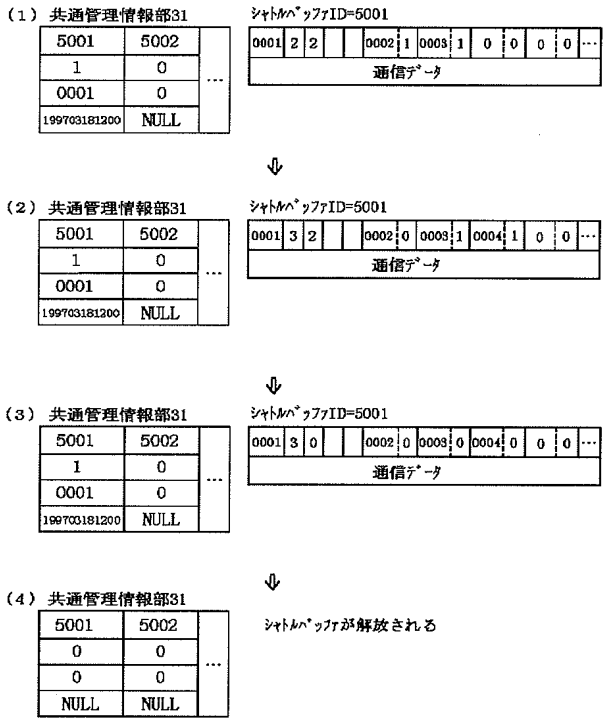
【図１】


開始
S11
共通管理情報部から使用可能なｼｬﾄﾙﾊﾞｯﾌｧIDを検索する
S12
ｼｬﾄﾙﾊﾞｯﾌｧID=nは使用可能?
N
Y
S13
共通管理情報部のｼｬﾄﾙﾊﾞｯﾌｧID=nの使用中ﾌﾗｸﾞを「使用中」にする
ｵｰﾅﾌﾟﾛｾｽIDを設定する
取得日時を設定する
S14
ｼｬﾄﾙﾊﾞｯﾌｧの領域を取得する
ｼｬﾄﾙﾊﾞｯﾌｧの
ｵｰﾅﾌﾟﾛｾｽIDを設定する
参照ﾌﾟﾛｾｽ数を設定する
未参照ﾌﾟﾛｾｽ数を設定する
参照ﾌﾟﾛｾｽIDを設定する
参照済みﾌﾗｸﾞを「未参照」に設定する
S15
ｼｬﾄﾙﾊﾞｯﾌｧの通信ﾃﾞｰﾀ部に通信ﾃﾞｰﾀを設定する
S16
ｿｹｯﾄ通信により通知先ﾌﾟﾛｾｽにｼｬﾄﾙﾊﾞｯﾌｧIDを通知する
終了

【図２】


開始
S21
ｿｹｯﾄ通信により通知されたIDによりｼｬﾄﾙﾊﾞｯﾌｧを参照する
S22
ｼｬﾄﾙﾊﾞｯﾌｧの参照するﾌﾟﾛｾｽIDに記述されている以外のﾌﾟﾛｾｽに通信ﾃﾞｰﾀを通知する?
N
Y
S23
ｼｬﾄﾙﾊﾞｯﾌｧの
参照ﾌﾟﾛｾｽ数を再設定する
未参照ﾌﾟﾛｾｽ数を再設定する
参照ﾌﾟﾛｾｽIDを設定する
参照済みﾌﾗｸﾞを「未参照」にする
ｿｹｯﾄ通信により通信先ﾌﾟﾛｾｽにｼｬﾄﾙﾊﾞｯﾌｧIDを通知する
S24
ｼｬﾄﾙﾊﾞｯﾌｧの自ﾌﾟﾛｾｽの参照済みﾌﾗｸﾞを「未参照」にする
未参照ﾌﾟﾛｾｽ数を-1する
S25
ｼｬﾄﾙﾊﾞｯﾌｧの未参照ﾌﾟﾛｾｽ数=0 ?
S26
ｼｬﾄﾙﾊﾞｯﾌｧID=nを解放する
S27
共通管理情報のｼｬﾄﾙﾊﾞｯﾌｧID=nの使用中ﾌﾗｸﾞを「未使用」にする
ｵｰﾅﾌﾟﾛｾｽIDをｸﾘｱする
取得日付をｸﾘｱする
終了

【図４】


（１）共通管理情報部31
5001
5002
1
0
0001
0
199703181200
NULL
シャドウバッファID=5001
0001 2 2 0002 1 0003 1 0 0 0 0
通信データ
（２）共通管理情報部31
5001
5002
1
0
0001
0
199703181200
NULL
シャドウバッファID=5001
0001 3 2 0002 0 0003 1 0004 1 0 0
通信データ
（３）共通管理情報部31
5001
5002
1
0
0001
0
199703181200
NULL
シャドウバッファID=5001
0001 3 0 0002 0 0003 0 0004 0 0 0
通信データ
（４）共通管理情報部31
5001
5002
0
0
0
0
NULL
NULL
シャドウバッファが解放される